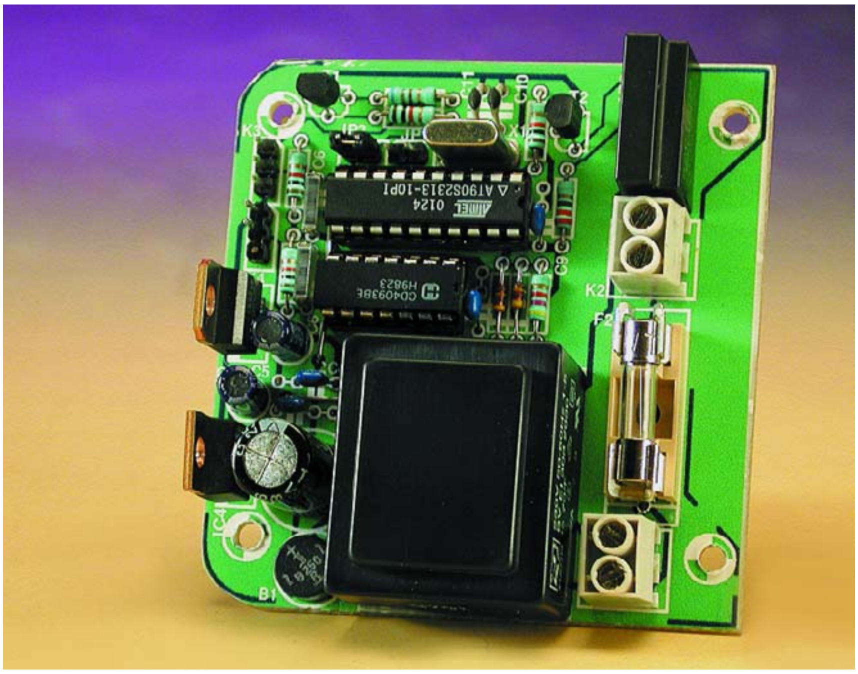
Night Light Control

using the AT90S2313

Design by V. Schmidt

The special feature of this night light is that it has a switch-off delay. This is achieved using a microcontroller with an integrated 16 bit counter.

This night light can be switched on and off as normal, but it also includes a timer which can automatically turn the light off after a preset time. The light can be fitted with two (or more) buttons, one of which can be situated 'locally' to the light while the other (or others) can be some distance away. The local button can sport LEDs which not only make it easier to find it in the darkness, but which also indicate the status of the night light controller.



The circuit

Because the circuit uses a microcontroller, the number of components is kept down to a reasonable level. The circuit in **Figure 1** shows the AVR microcontroller type AT90S2313, which drives a solid state relay type S201S02 (IC5) from PD0 via T2. This relay is an electrically isolated switch connected in series with K1 (the mains supply) and K2 (the lamp). The relay includes a zero-crossing detector which turns on the connected load as the mains voltage goes through zero. The microcontroller and the rest of the circuit are powered from regulated DC supplies at +5 V (IC4) and +12 V (IC3), also, of course, isolated from the mains.

The two buttons are connected via K3 and K4 and drive port pin PB0 via a diode-OR configuration. In the quiescent state, when neither button is pressed, R2 and R3 hold the respective inputs of the inverting Schmitt triggers low, and so the outputs will be high. D4 and D5 allow the output level to be different from the input level, and R4 ensures that the microcontroller sees a high-level input voltage of 5 V. The gates are therefore also operating as level shifters. So why go to the trouble of using two different supply voltages? The reason is simple: with a possibly rather long cable between the controller circuit board and the buttons, a higher supply voltage greatly increases the noise immunity of the circuit at the cost of just one voltage regulator and three capacitors.

If one of the buttons is pressed, the level at the input to the corresponding gate goes high, and so its output goes low. Taking into account the forward voltage drop of the diode, the level at the input to the microcontroller will be around 0.9 V, which it will safely interpret as low. The signal is not debounced here: that job is done in software in the AVR microcontroller.

Transistor T1 is controlled via output PD1 or PD2, depending on which of jumpers JP1 and JP2 is fitted, and drives the LEDs next to button S1. If JP1 is fitted, the LEDs are normally illuminated continuously and flash when the sleep timer is running. If JP2 is fitted the LEDs are normally off but flash when the timer is running. The microcontroller is clocked at 4.9152 MHz. This unusual value is required to allow an internal signal at 10 Hz to be generated.

The software

So much for the hardware: now to the software. This was created and tested using the Atmel STK-500 starter kit. The WAVRASM assembler was used along with AVR Studio 3.2 for testing and downloading the code into the microcontroller.

The night light controller program is available for free download from the *Elektor Electronics* website in assembler and in compiled form. It is divided into six sections which we shall now discuss.

Initialisation (INI)

This is where the parameters are set that configure the functions of the internal components of the microcontroller. First the interrupt vector for Timer 1, running in compare mode, is set to the address of the interrupt code (label ‘ini’). The compare interrupt for Timer 1 is now enabled by setting bit OCIE1A in register TIMSK. The prescaler for Timer 1 is set to divide by 1024 in register TCCR1B and the CTC option is enabled. When the preset value in registers OCR1AL and OCR1AH is reached, the timer is cleared and the process begins anew. This maximum value is now set: 30H in OCR1AL and 00H in OCR1AH. This corresponds to 48 in decimal. With the prescaler, and with a crystal frequency of 4.9152 MHz, the interrupt routine is called every 10 ms. Next the data direction registers for ports B and D are configured. Port B operates as an input, port D as an output. Finally the I bit is set in SREG to enable interrupts.

Main program

The main program consists of just four instructions, calling subroutines INPUT, LOGICOUT and OUTPUT. Then the main loop repeats from the beginning.

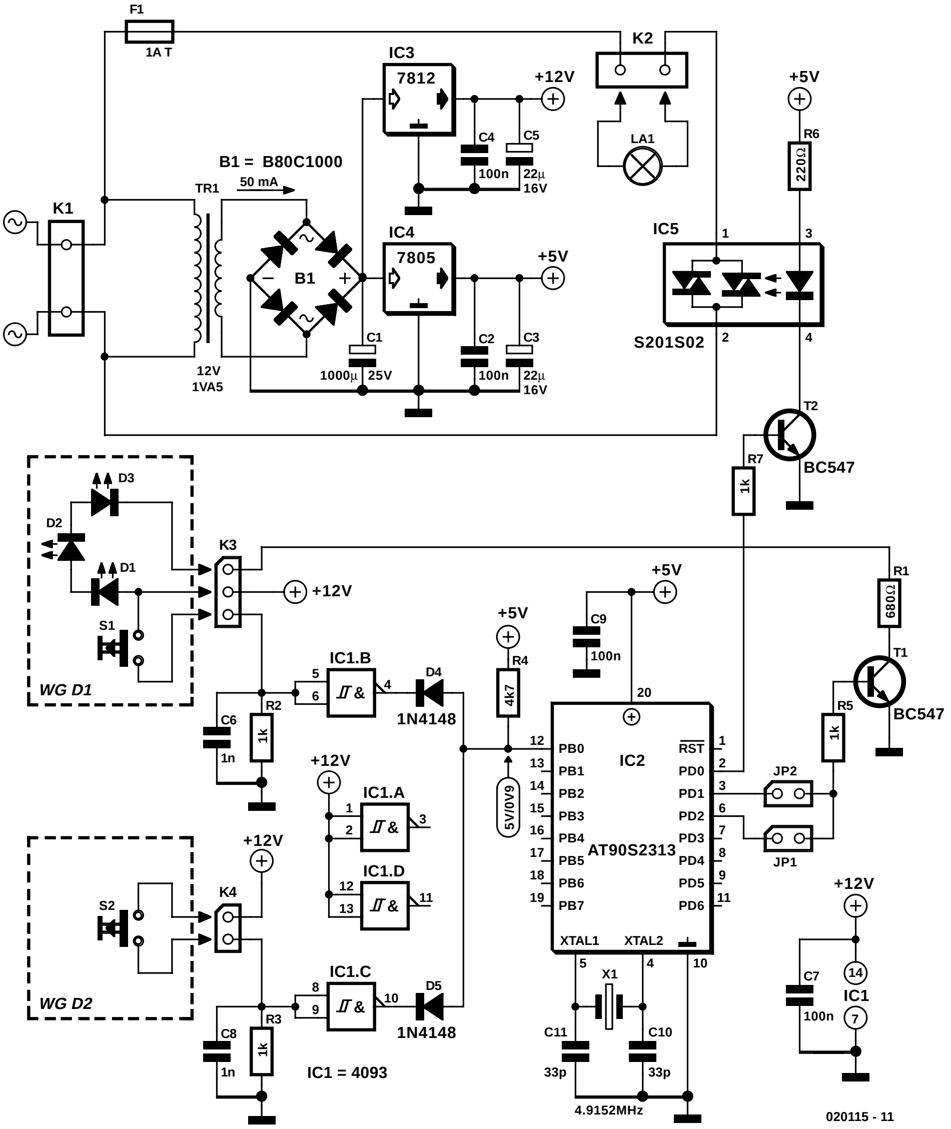


Figure 1. Circuit diagram of the night light controller with remote buttons.

Interrupt-driven timer routine (SUB TIMER)

The interrupt routine creates a basic clock period of 10 ms, which is used by DELAY for debouncing the buttons and incrementing register TAKT10 (‘clock 10’), from which all time-dependent quantities are derived. TAKT (‘clock’), which is used for measuring the duration of a button press, is incremented every 100 ms, when register TAKT10 reaches 10 (decimal). TAKT10 is then reset to zero and the process repeats from the beginning.

Next a one-second clock is generated for the ‘status display’ in subroutine OUTPUT, and a ten-second clock for the output timer (OUTC) in subroutine LOGICOUT. Flags F1SEC and F10SEC in STEUER (‘control’) are set according on the state of registers SEC1 and SEC10. Flag F1SEC is 1 for 500 ms and then 0 for a further 500 ms. Flag F10SEC is simply set every 10 s and subsequently cleared in subroutine LOGICOUT.

SUB INPUT

Subroutine INPUT is activated when port pin PB0 is taken low. When a button is pressed, the falling edge on the input starts DELAY for debouncing. On return from DELAY the port pin is tested again to verify that it is still zero. If this is the case the contents of register TAKT are copied into register START and flag FKENN (‘edge detect’) is set. Otherwise control returns to the main program. When the button is released the contents of register TAKT are copied into register STOP: this allows the period for which the button was pressed to be determined. Overflow is prevented by comparing registers START and STOP. If the period for which the button was pressed is less than 1.5 s (a count of 0F hex) flag FKL15 (‘less than 15’) is set; if it is greater, flag FGR15 is set. Then, in both cases, the edge detection flag FKENN is cleared.

SUB LOGICOUT

Subroutine LOGICOUT contains the logic for generating the output and controlling the sleep timer in register OUTC. First a check is made to see whether flag FKL15 or FGR15 is set. In the first case flag OUT in register STEUER is toggled (and the lamp will then be either turned on or turned off). In the second case flags OUT and OTIML are set. OTIML is the start condition for the sleep timer in reg-

AT90S2313 register settings

Timer 1 settings

Timer interrupt: TIMSK

TOIE1	OCIE1A				TICIE1		TOIE0	HEX
0	1	0	0	0	0	0	0	40

(Timer/Counter 1 compare match interrupt enabled. An interrupt is generated when the counter value becomes equal to the compare register contents.)

CTC and prescaler: TCCR1B

ICNC1	ICES1			CTC1	CS12	CS11	CS10	HEX
0	0	0	0	1	1	0	1	0D

(Clear Timer/Counter 1 to zero when count value equals compare register; set prescaler to divide by 1024 (CS12 to CS10).)

Compare register

OCR1AH	00h
OCR1AL	30h

0030H = 48 decimal

(Timer 1 operates as a clock generator and generates an interrupt every 10 ms. The crystal frequency of 4915200 Hz is divided down to 100 Hz by the prescaler and Timer 1.)

Stack pointer setting

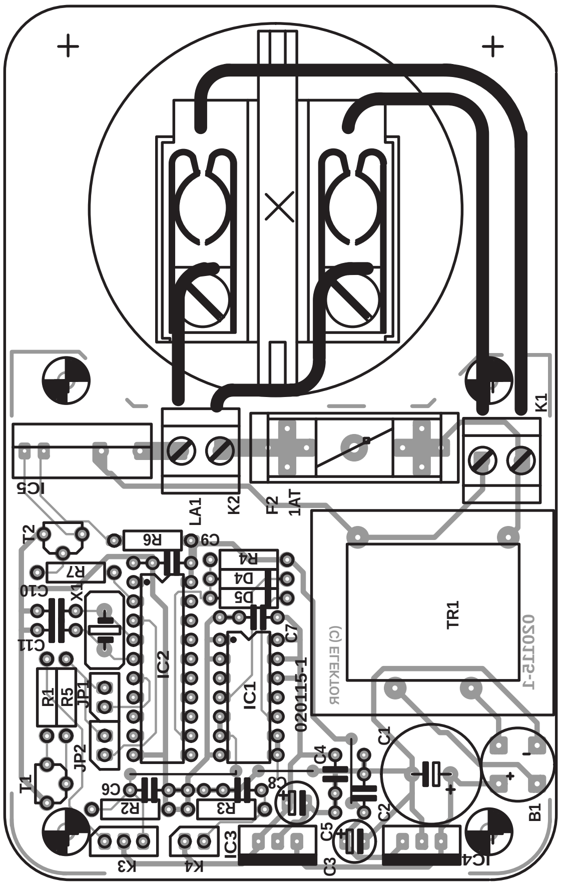
SPL = DFh

Port control settings

Register DDRD	FFh: port D configured as output
Register DDRB	00h: port B configured as input

SREG setting

Set I bit to enable interrupts.



ister OUTC. This register is incremented every 10 s until it reaches the constant value OUTIM (180 decimal). Then both OUT and OTIML are cleared and the program returns to its initial state.

SUB OUTPUT

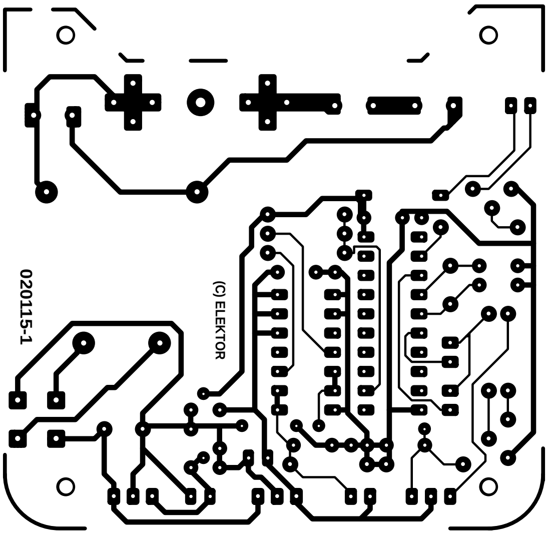
OUTPUT controls the output on port D. The action of this subroutine depends on the state of flag OUT. If it is set, it jumps to label OUTPUT1. There PD0 is switched high, and then the state of flag OUTIML is used to control further execution. If this flag is clear, subroutine OUTPUT terminates with a return to the main program. If, however, it is set, the succeeding assembler instructions ensure that 1 Hz clock signals are output on port pins PD1 and PD2, 180 ° out of phase with one another.

If flag OUT is not set port pins PD0 to PD2 are set to their default levels (PD0 and PD1 to zero, PD2 to one) and the subroutine ends.

SUB DELAY

Subroutine DELAY debounces the buttons using the 10 ms counter TAKT10. When the routine is called the counter's state is copied into register DELAY and the value incremented by 2, representing 20 ms. If this results in a value greater than ten, ten is subtracted from DELAY. This is important, since counter TAKT10 is reset to zero when it

Figure 2. The night light controller is built in an enclosure with a mains plug and socket (board available from The PCBShop).



COMPONENTS LIST

Resistors:
R1 = 680Ω
R2,R3,R5,R7 = 1kΩ
R4 = 4kΩ7
R6 = 220Ω

Capacitors:
C1 = 1000µF 25V radial
C2,C4,C7,C9 = 100nF
C3,C5 = 22µF 16V radial
C6,C8 = 1nF
C10,C11 = 33pF

Semiconductors:
B1 = B80C1000 bridge rectifier in round case (80V piv, 1 A)
D1,D2,D3 = LED
D4,D5 = 1N4148
T1,T2 = BC547
IC1 = 4093
IC2 = AT90S2313-10PC (order code **020115-41**)
IC3 = 7812
IC4 = 7805
IC5 = S201S02 or S201SE2

Miscellaneous:
F1 = fuse, 1A(T) (time lag) with PCB mount holder
JP1,JP2 = 2-way jumper
K1,K2 = 2-way PCB terminal block, lead pitch 5mm
K3 = 3-way pinheader or solder pins
K4 = 2-way pinheader or solder pins
S1,S2 = pushbutton, 2 make contacts (see text)
TR1 = mains transformer, 12V 1.5VA (e.g, Hahn BV EI 302 2022)
X1 = 4.9152MHz quartz crystal
Disk, hex and source code, order code **020115-11** or free download

reaches ten. The following loop is then executed until the contents of DELAY and TAKT10 become equal. Then the subroutine returns.

Mechanical construction

The circuit is built inside an enclosure with a mains plug and socket. The tiny circuit board in **Figure 2** may have to be filed down slightly to fit inside the enclosure of your choice. There should be no difficulties with fitting the components and the circuit should work first time as long as the components are properly soldered and the three wire links are not overlooked.

The button (or buttons) can if desired be mounted separately in their own enclosures. The cable is taken through the underside of the enclosure (not forgetting a strain relief) and connected to terminals K3 and K4. It is sensible, however, to fit S1 along with its LEDs in the main enclosure.

(020115-1)

Free Downloads

- Night light controller program in assembler and in compiled form (source code and HEX file).
  File number: **020115-11.zip**
- Printed circuit board layout in PDF format.
  File number: **020115-1.zip**

www.elektor-electronics.co.uk/dl/dl.htm
(select month of publication)